Math 58B - Introduction to Biostatistics

Jo Hardin

Spring 2017

Example R code / analysis for housing data

```
library(GGally)
library(ggplot2)
library(dplyr)
house = read.table("http://www.rossmanchance.com/iscam2/data/housing.txt", header=TRUE, sep="\t")
names(house)
```

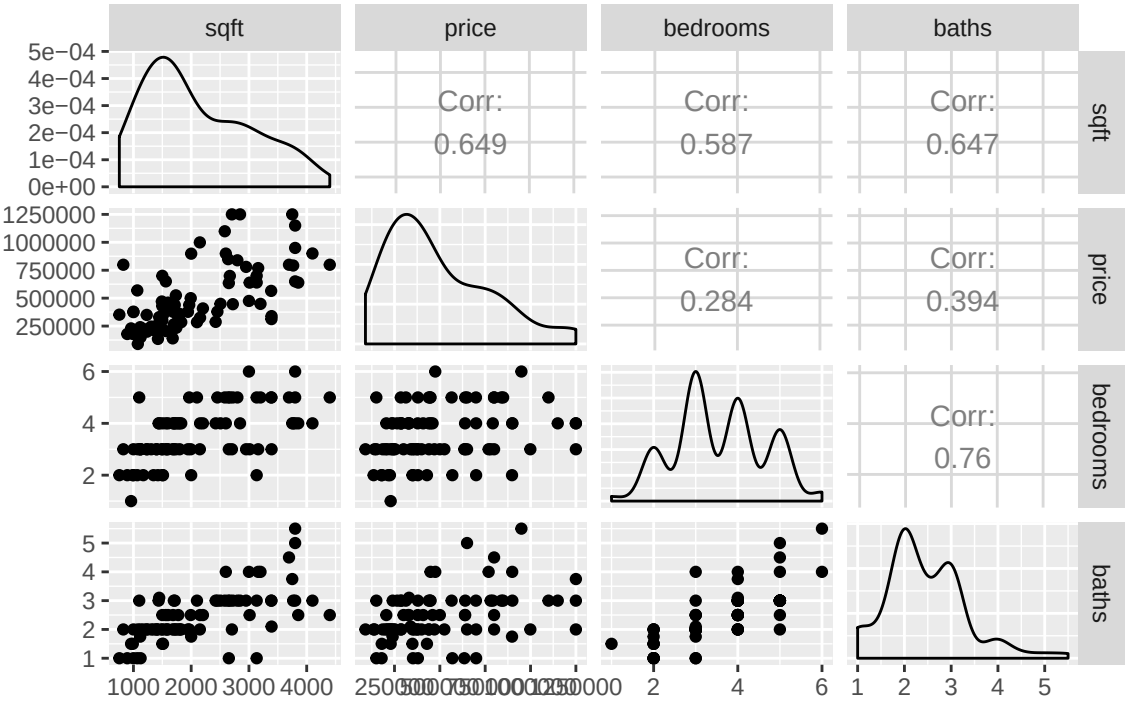
```
## [1] "sqft"     "price"    "City"     "bedrooms" "baths"
```

Goals

1. Understand how to run a regression with multiple variables.
2. Understand how to check residual plots for normal assumptions.
3. Understand how to find the "best" model (acknowledging that there is never such thing as "best").

Descriptive Statistics

```
ggpairs(house, columns = c(1,2,4,5))
```



Run a linear model trying to predict price

```
mod.sqft <- lm(price~sqft, data = house)
summary(mod.sqft)
```

```
##
## Call:
## lm(formula = price ~ sqft, data = house)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -439654 -144256  -52040   97373  636508
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 65930.31   60993.62   1.081    0.283
## sqft          202.43      26.39   7.670 3.35e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 222100 on 81 degrees of freedom
## Multiple R-squared:  0.4207, Adjusted R-squared:  0.4136
## F-statistic: 58.83 on 1 and 81 DF,  p-value: 3.349e-11
```

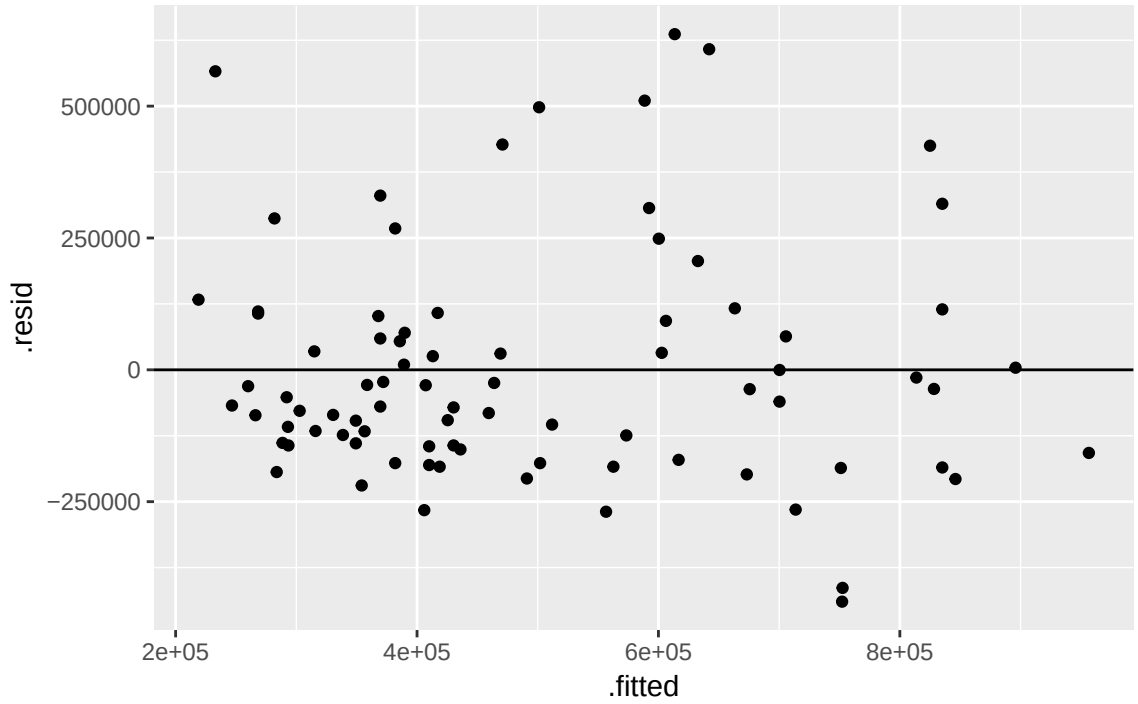
```
mod.price <- lm(price ~ bedrooms, data=house)
summary(mod.price)
```

```
##
## Call:
## lm(formula = price ~ bedrooms, data = house)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -454935 -206553  -76206  190930  798794
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   220612     107208   2.058  0.04283 *
## bedrooms       76865      28802   2.669  0.00919 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 279800 on 81 degrees of freedom
## Multiple R-squared:  0.08082,    Adjusted R-squared:  0.06947
## F-statistic: 7.122 on 1 and 81 DF,  p-value: 0.009195
```

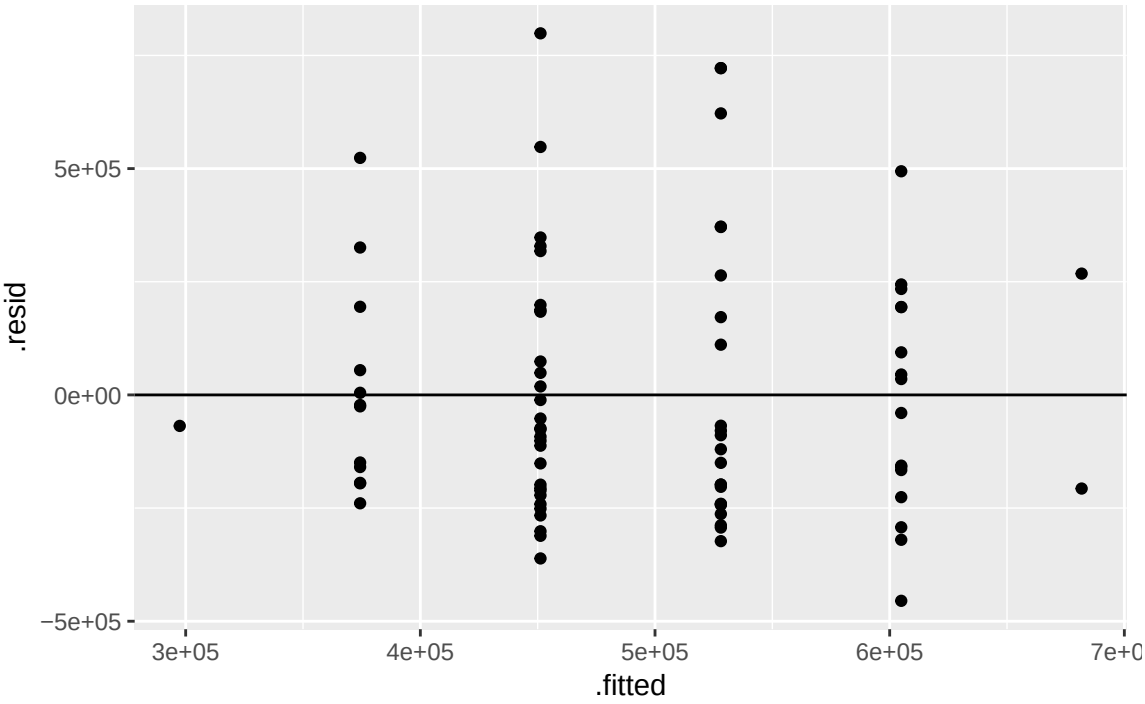
The p-values for both explanatory variables (sqft and bedrooms) are significant. Sqft seems more significant, and indeed, the first model has a higher R^2 - that is, a higher proportion of the variability in price is explained by sqft (42.07%) than by number of bedrooms (8.08%).

However, it is important for us to ask whether either of the relationships actually fit the technical conditions of the linear regression model. We can see from the pairs plots that the relationships look **L**inear, we'll assume the variables were collected **I**ndependently, but the **N**ormality and the **E**quality of the error structure we can check using residual plots.

```
ggplot(mod.sqft, aes(.fitted, .resid)) + geom_point() + geom_hline(yintercept=0)
```



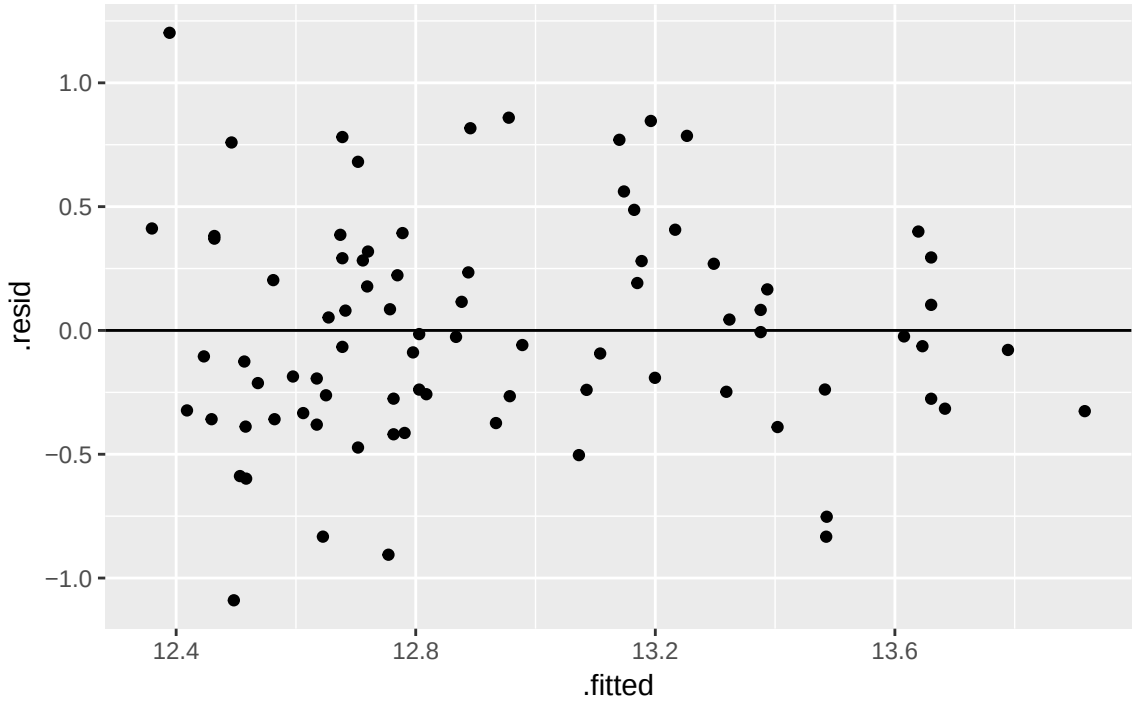

```
ggplot(mod.price, aes(.fitted, .resid)) + geom_point() + geom_hline(yintercept=0)
```



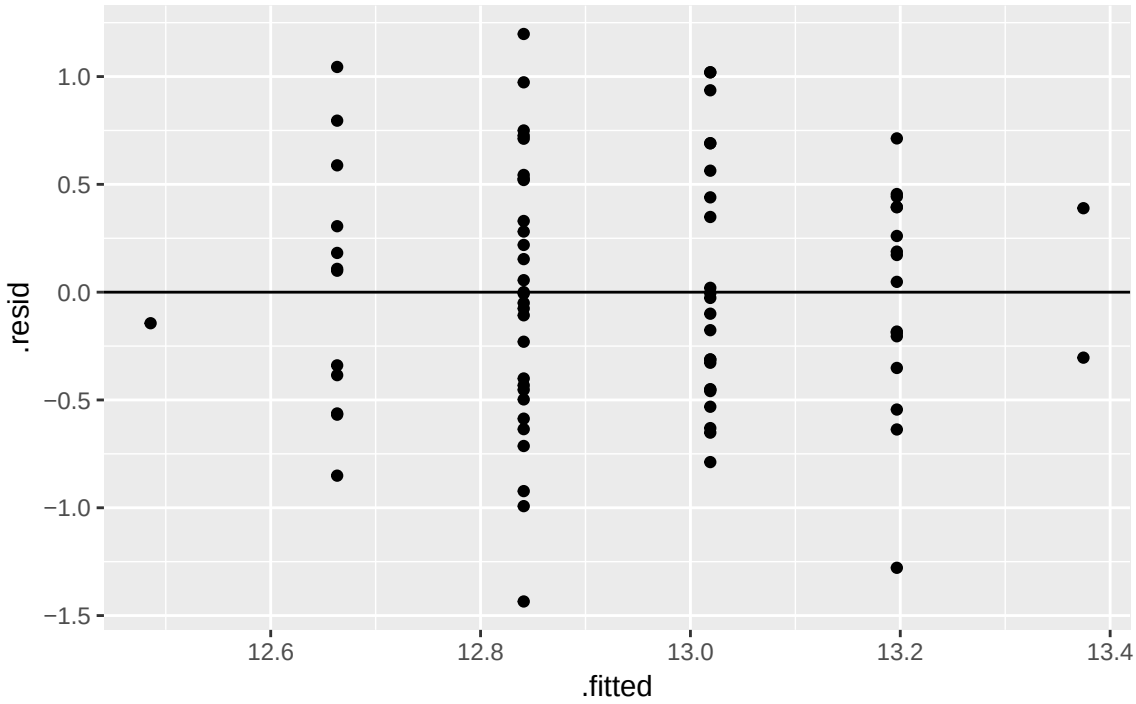
For both of the plots, it seems like the residuals have higher variability for positive residuals. Additionally, it seems that the variability of the residuals increases for larger fitted observations.

A natural log transformation should fix both of these problems.

```
mod.lnsqft <- lm(log(price)~sqft, data = house)
mod.lnprice <- lm(log(price) ~ bedrooms, data=house)
ggplot(mod.lnsqft, aes(.fitted, .resid)) + geom_point() + geom_hline(yintercept=0)
```




```
ggplot(mod.lnprice, aes(.fitted, .resid)) + geom_point() + geom_hline(yintercept=0)
```



Though no residual plot will ever look perfect, these residual plots seem to fit the technical conditions of the model better than the untransformed data.

Combining variables.

We'll stick with the transformed data. What happens when we try to predict price (log(price), here) using BOTH sqft and bedrooms?

```
summary(lm(log(price) ~ sqft + bedrooms, data=house))
```

```
##
## Call:
## lm(formula = log(price) ~ sqft + bedrooms, data = house)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.0814 -0.2776 -0.0530  0.2680  1.2208
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.216e+01  1.736e-01  70.092  < 2e-16 ***
## sqft         4.683e-04  6.603e-05   7.092 4.73e-10 ***
## bedrooms    -6.029e-02  5.720e-02  -1.054    0.295
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4499 on 80 degrees of freedom
## Multiple R-squared:  0.4484, Adjusted R-squared:  0.4346
## F-statistic: 32.52 on 2 and 80 DF,  p-value: 4.623e-11
```

```
summary(lm(log(price) ~ sqft + baths, data=house))
```

```
##
## Call:
## lm(formula = log(price) ~ sqft + baths, data = house)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.08265 -0.30018 -0.05004  0.27759  1.21507
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.208e+01  1.510e-01  80.006  < 2e-16 ***
## sqft         4.504e-04  7.046e-05   6.392 1.02e-08 ***
## baths       -3.770e-02  7.460e-02  -0.505    0.615
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4523 on 80 degrees of freedom
## Multiple R-squared:  0.4425, Adjusted R-squared:  0.4286
## F-statistic: 31.75 on 2 and 80 DF,  p-value: 7.066e-11
```

```
summary(lm(log(price) ~ sqft + bedrooms + baths, data=house))
```

```
##
## Call:
## lm(formula = log(price) ~ sqft + bedrooms + baths, data = house)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.08355 -0.28314 -0.04766  0.26424  1.21742
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.216e+01  1.751e-01  69.453  < 2e-16 ***
## sqft         4.634e-04  7.187e-05   6.448 8.32e-09 ***
## bedrooms    -6.827e-02  7.302e-02  -0.935    0.353
## baths        1.681e-02  9.472e-02   0.177    0.860
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4527 on 79 degrees of freedom
## Multiple R-squared:  0.4486, Adjusted R-squared:  0.4277
## F-statistic: 21.43 on 3 and 79 DF,  p-value: 2.982e-10
```

Although the R^2 value went up (44.84% of variability in log price is explained by sqft and bedrooms), the p-value on bedrooms isn't significant. The p-value here can be interpreted as a hypothesis test on the slope coefficient given the other variables in the model.

0.353 = P(a slope of -.06827 or more extreme *if sqft is in the model* and there is no relationship between bedrooms and price)

Our output says that once we have sqft in the model, we don't actually need to know anything about the number of bedrooms (even though bedrooms was a significant predictor on its own).

The final model will be run on log(price) using only sqft. Note that the coefficients and the R^2 values change slightly (from the original analysis) because the response variable is logged.

```
summary(mod.lnsqft)
```

```
##
## Call:
## lm(formula = log(price) ~ sqft, data = house)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.08988 -0.29591 -0.05899  0.28717  1.20206
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.204e+01  1.236e-01   97.36  < 2e-16 ***
## sqft        4.274e-04  5.349e-05    7.99 7.87e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4502 on 81 degrees of freedom
## Multiple R-squared:  0.4407, Adjusted R-squared:  0.4338
## F-statistic: 63.83 on 1 and 81 DF,  p-value: 7.874e-12
```

Prediction

As with the prediction intervals we had when we had a single sample, we can now create intervals for either an average (a confidence interval) of an individual (a prediction interval).

Confidence interval:

```
predict(mod.lnsqft, newdata=data.frame(sqft=2000), interval="confidence")
```

```
##        fit      lwr      upr
## 1 12.89125 12.79211 12.99038
```

I am 95% confident that the true average log price for a 2000 sqft home is between 12.79 log$ and 12.99 log$. Back transforming can be a little tricky.

Prediction interval:

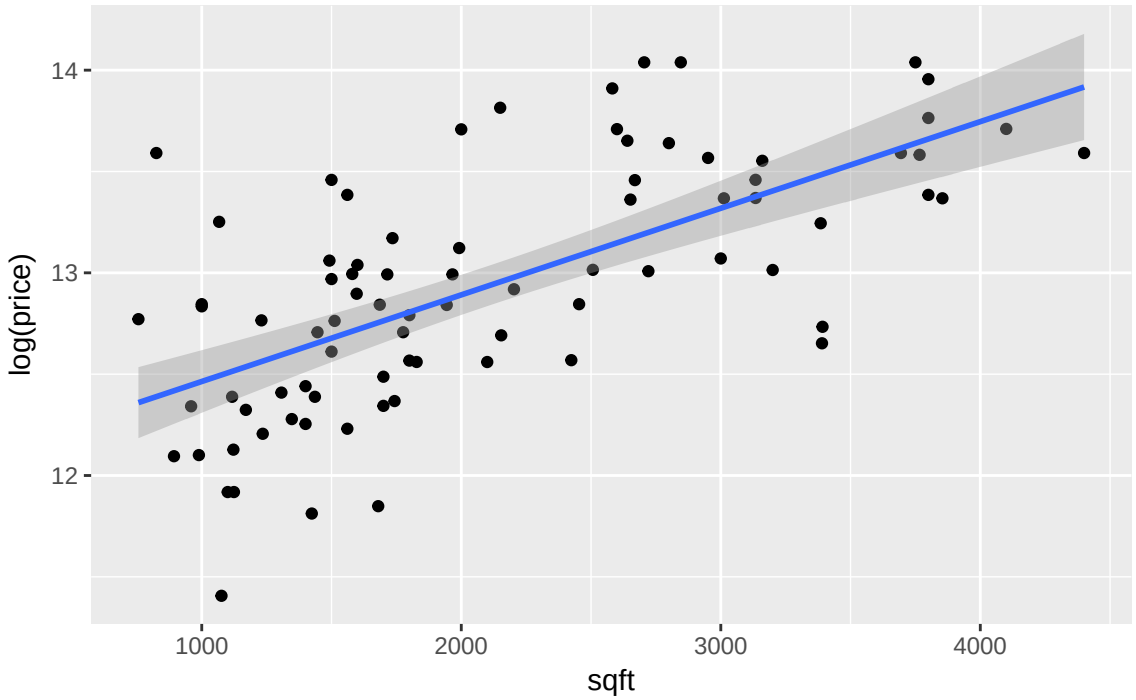
```
predict(mod.lnsqft, newdata=data.frame(sqft=2000), interval="prediction")
```

```
##        fit      lwr      upr
## 1 12.89125 11.98994 13.79255
```

I am 95% of homes with 2000 sqft are between 11.99 log$ and 13.79 log$. Now back transforming is easy (because there are no averages), so 95% of homes with 2000 sqft are between $161,126 and $977,301.

Plotting the confidence bounds on a scatterplot

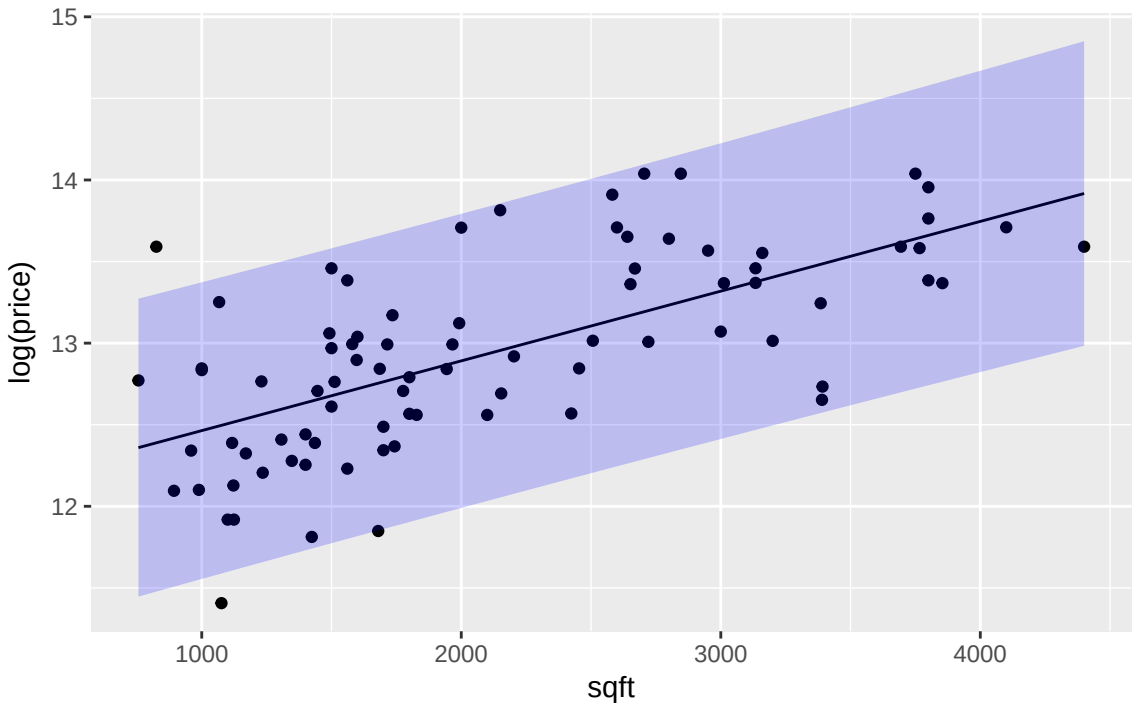
```
ggplot(house, aes(x=sqft, y=log(price))) + geom_point() + geom_smooth(method="lm", level=.95)
```



```
housePred <- cbind(house,predict(mod.lnsqft, interval="prediction") )
names(housePred)
```

```
## [1] "sqft"     "price"    "City"     "bedrooms" "baths"    "fit"
## [7] "lwr"      "upr"
```

```
ggplot(housePred, aes(x=sqft)) + geom_point(aes(y=log(price))) + geom_line(aes(y=fit)) +
  geom_ribbon(aes(ymin=lwr, ymax=upr), fill="blue", alpha=0.2)
```



Predicting with more than one explanatory variable

The predict function still works to give estimates of the average value and the predicted individual values, but the plot is now much harder to draw because with three variables, we would need a 4-d plot.

```
sqftbedbathlm = lm(log(price)~sqft + bedrooms + baths, data=house)
predict(sqftbedbathlm, newdata=data.frame(sqft=2000, bedrooms=3, baths=2), interval="confidence", level=
```

```
##        fit      lwr      upr
## 1 12.91816 12.80085 13.03548
```

```
predict(sqftbedbathlm, newdata=data.frame(sqft=2000, bedrooms=3, baths=2), interval="prediction", level=
```

```
##        fit      lwr     upr
## 1 12.91816 12.00953 13.8268
```

Again, it is hard to back transform the prediction for the average (we end up thinking about it as a median), but we can back transform the interval of individual prices. 95% of homes with 2000sqft, 3 bedrooms, and 2 baths cost between $164,312 and $1,011,356.